ELSEVIER

Signal Processing: *Image Communication* 00 (2000) 000–000

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.nl/locate/image

# A texture descriptor for browsing and similarity retrieval

B.S. Manjunath*, P. Wu, S. Newsam, H.D. Shin[1]

*Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560, USA*

**Abstract**

Image texture is useful in image browsing, search and retrieval. A texture descriptor based on a multiresolution decomposition using Gabor wavelets is proposed. The descriptor consists of two parts: a perceptual browsing component (PBC) and a similarity retrieval component (SRC). The extraction methods of both PBC and SRC are based on a multiresolution decomposition using Gabor wavelets. PBC provides a quantitative characterization of the texture's structuredness and directionality for browsing application, and the SRC characterizes the distribution of texture energy in different subbands, and supports similarity retrieval. This representation is quite robust to illumination variations and compares favorably with other texture descriptors for similarity retrieval. Experimental results are provided. 

*Keywords:* Texture descriptor; Similarity retrieval; Perceptual browsing; Multiresolution decomposition

## 1. Introduction

The recent advances in digital imaging and computing technology have resulted in a rapid accumulation of digital media in the personal computing and entertainment industry. In addition, large collections of such data already exist in many scientific application domains such as the geographic information systems (GIS) and medical imaging. Managing large collections of multimedia data requires development of new tools and technologies. This is evident in the current MPEG-7 standardization effort whose objective is *to provide a set of standardized tools to describe the multimedia content* [9,15,16].

At the core of the MPEG-7 is a set of *descriptors* for audio-visual content. In [16] a descriptor is defined as a representation of a feature. A descriptor defines the syntax and semantics of the feature representation. Examples of low-level visual features include color, shape, motion, and texture. This paper describes a texture feature descriptor that is being proposed to the MPEG-7 standard [18]. Key functionalities supported by this descriptor include image browsing and similarity-based retrieval.

Image texture has emerged as an important visual primitive to search and browse through large collections of similar looking patterns. An image can be considered as a mosaic of textures and texture features associated with the regions can be used to index the image data. For instance, a user browsing an aerial image database may want to

* Corresponding author.

*E-mail addresses:* manj@iplab.ece.ucsb.edu, manj@surya.ece.ucsb.edu (B.S. Manjunath), leng@iplab.ece.ucsb.edu (P. Wu), snewsam@iplab.ece.ucsb.edu (S. Newsam), hdshin@iplab.ece.ucsb.edu (H.D. Shin).

[1] Dr. Shin is currently a visiting researcher at UCSB, on leave from Samsung Electronics.

0923-5965/00/$ - see front matter 
PII: S0923-5965(00)00016-3

identify all parking lots in the image collection. A parking lot with cars parked at regular intervals is an excellent example of a textured pattern when viewed from a distance, such as in an airphoto. Similarly, agricultural areas and vegetation patches are other examples of textures commonly found in aerial and satellite imagery. Examples of queries that could be supported in this context could include "Retrieve all Landsat images of Santa Barbara which have less than 20% cloud cover" or "Find a vegetation patch that looks like this region". To support image retrieval or browsing, an effective representation of textures is required.

One of the widely used representations of textures is the texture feature proposed in [17] and its improved version in [6]. The texture feature used in [17,6] is based, to some extent, on models of human texture perception. More recently, several random-field-based texture models [10,14] and multiscale filtering methods [3,13] have been studied. Use of texture for content-based retrieval has been explored by several researchers [6,11,12]. Among these, features computed from Gabor filtered images appear quite promising. A comprehensive evaluation of using Gabor features can be found in [11,13]. More recent evaluation and comparison using other texture features also support the observation that the orientation and scale-selective Gabor filtered images capture relevant texture properties for applications such as image retrieval [8].

The proposed texture descriptor is based on Gabor filtering [11,13]. The descriptor has two parts: The first part relates to a perceptual characterization of texture in terms of structuredness, directionality and coarseness (scale). This representation is useful for browsing type applications and coarse classification of textures. We call this part the perceptual browsing component (PBC). The second part provides a quantitative description that can be used for accurate search and retrieval. This is referred to as the similarity retrieval component (SRC). The SRC component is described in detail in an earlier paper [13]. Both of the components are derived from a multiresolution Gabor filtering. Key features of this descriptor are

- It captures both the high-level perceptual characterization (in terms of directionality, structuredness, and coarseness of a texture), as well as a robust quantitative characterization at multiple scales and orientations.
- Feature extraction is simple, involving image convolutions with a set of masks. The filters are based on a 2-D Gabor wavelet decomposition. Image convolutions can be efficiently implemented in hardware and software.
- Multiple applications can be supported by the descriptor. For example, by using PBC, browsing of image database could be performed (e.g., *show textures that are structured and are oriented at 90°*). The SRC can be used for query by example type applications wherein similarity retrieval is needed.

The paper is organized as follows. The next section provides a brief introduction to Gabor filters. Computing the PBC is described in Section 3 and Section 4 details SRC computation. Experimental results are provided in Section 5. Section 6 concludes with discussions.

## 2. Gabor filter bank [13]

The use of Gabor filters in extracting texture descriptors is motivated by several factors. The Gabor representation has been shown to be optimal in the sense of minimizing the joint two-dimensional uncertainty in space and frequency [4]. These filters can be considered as orientation and scale tunable edge and line detectors, and the statistics of these micro features can be used to characterize the underlying texture.

A two-dimensional Gabor function and its Fourier transform can be written as

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right) + 2\pi jWx\right], \quad (1)$$

$$G(u, v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\}, \quad (2)$$

where $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$. A class of self-similar functions, refered to as the Gabor wavelets, is now considered. Let $g(x, y)$ be the

mother wavelet. Then a self-similar filter dictionary can be obtained by appropriate dilations and translations of $g(x, y)$ through the generation function [13]:

$$g_{mn}(x, y) = a^{-m} g(x', y'), \quad a > 1, \quad m, n = \text{integer}$$

$$x' = a^{-m}(x \cos\theta + y \sin\theta) \quad \text{and}$$

$$y' = a^{-m}(-x \sin\theta + y \cos\theta), \tag{3}$$

where $\theta = n\pi/K$ and $K$ is the total number of orientations. The scale factor $a^{-m}$ in (3) is meant to ensure that the energy is independent of $m$. This set of functions form a non-orthogonal basis of functions for the multiresolution decomposition [13].

The non-orthogonality of the Gabor wavelets implies that there is redundant information in the filtered images, and the following strategy is used to reduce this redundancy. Let $U_l$ and $U_h$ denote the lower and upper center frequencies of interest. Let $K$ be the number of orientations and $S$ be the number of scales in the multiresolution decomposition. Then the design strategy is to ensure that the half-peak magnitude supports of the filter responses in the frequency spectrum touch each other as shown in Fig. 1. This results in the following formulas for computing the filter parameters $\sigma_u$ and $\sigma_v$ (and thus $\sigma_x$ and $\sigma_y$) [13].

$$a = (U_h/U_l)^{1/(S-1)}, \quad \sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}},$$

$$\sigma_v = \tan\left(\frac{\pi}{2k}\right)\left[U_h - 2\ln 2\left(\frac{\sigma_u^2}{U_h}\right)\right] \times \left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{U_h^2}\right]^{-1/2} \tag{4}$$

where $W = U_h$ and $m = 0, 1, \ldots, S-1$. In order to eliminate sensitivity of the filter response to absolute intensity values, the real (even) components of the 2-D Gabor filters are biased by adding a constant to make them zero mean (This can also be done by setting $G(0, 0)$ in (2) to zero.) Filtering the image $I(x, y)$ with $g_{mn}(x, y)$ results in

$$W_{mn}(x, y) = \int I(x, y) g^*_{mn}(x - x_1, y - y_1)\, dx_1\, dy_1, \tag{5}$$

where $*$ indicates the complex conjugate.

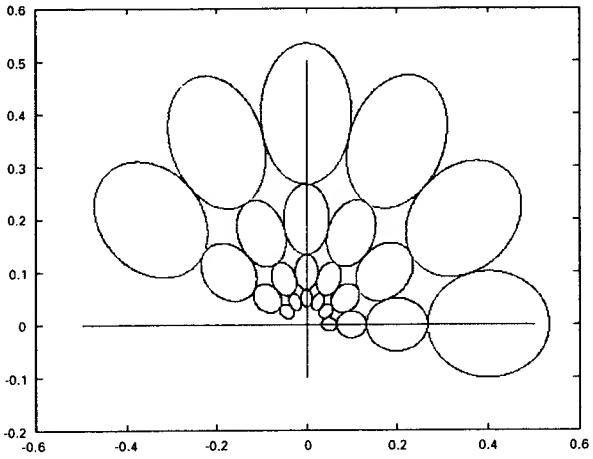



Fig. 1. The contours indicate the half-peak magnitude of the filter responses in the Gabor filter dictionary. The filter parameters used are $U_h = 0.04$, $U_l = 0.05$, $K = 6$ and $S = 4$ [6].

## 3. Perceptual browsing component (PBC)

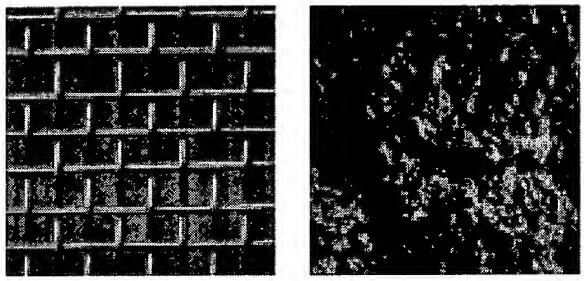
From the multiresolution decomposition, a given image is decomposed into a set of filtered images. Each of these images represents the image information at a certain scale and at a certain orientation. The PBC captures the regularity (or the lack of it) in the texture pattern. Its computation is based on the following observations:

- Structured textures usually consist of dominant periodic patterns.
- A periodic or repetitive pattern, if it exists, could be captured by the filtered images. This behavior is usually captured in more than one filtered output.
- The dominant scale and orientation information can also be captured by analyzing projections of the filtered images.

Based on the above observations, we propose the following format for the PBC:

$$PBC = [v_1\ v_2\ v_3\ v_4\ v_5]. \tag{6}$$

- *Regularity* ($v_1$): $v_1$ represents the degree of regularity or structuredness of the texture. A larger value of $v_1$ indicates a more regular pattern. Consider the two patterns in Fig. 2. Pattern Fig. 2(a) is intuitively more "regular" than Fig. 2(b),



(a) (b)

Fig. 2. Two examples of regularity of textures (a) regular pattern (b) irregular pattern.

and hence should have a larger $v_1$ compared to Fig. 2(b).

- *Directionality* $(v_2, v_3)$: These represent the two dominant orientations of the texture. The accuracy of computing these two components often depends on the level of regularity of the texture pattern. In our implementation, the orientation space is divided into 30° intervals.
- *Scale* $(v_4, v_5)$: These represent two dominant scales of the texture. Similar to directionality, the more structured the texture, the more robust the computation of these two components.

The PBC computation is a two step procedure. The first step is the analysis of each filtered output. The objective of this step is to determine the existence of a repetitive pattern. The second step is performed on all filtered outputs that are identified as having some kind of regularity.

### 3.1. Analysis of each filtered image and candidate selection

To identify if a filtered image is repetitive or not, the projections of each filtered image is computed and analyzed. The regular projections would be identified and further grouped to find dominant regularity of projections. The detail of the analysis is given below step by step.

*Projection*: For each filtered image, the projections along horizontal and vertical directions are computed. For an $N \times N$ image, the horizontal projection $P_H$ and vertical projection $P_V$ are defined as

$$P_H^{(mn)}(l) = \frac{1}{N}\sum_{k=1}^{N} W_{mn}(l,k) \quad \text{and}$$

$$P_V^{(mn)}(k) = \frac{1}{N}\sum_{l=1}^{N} W_{mn}(l,k), \tag{7}$$

where $l, k = 1, \ldots, N$, $W_{mn}(l,k)$ represents the $(m,n)$th filtered output. For simplicity in notation, we drop the index $(m,n)$ and the subscripts (H and V) in the following discussion.

*Autocorrelation*: Consider now a projection $P(l)$. The normalized autocorrelation function (NAC) is defined as

$$NAC(k) = \frac{\sum_{m=k}^{N-1} P(m-k)P(m)}{\sqrt{\sum_{m=k}^{N-1} P^2(m-k)\sum_{m=k}^{N-1} P^2(m)}}. \tag{8}$$

Fig. 3 shows the horizontal projections of texture pattern (a) in Fig. 2.

*Peak detection*: The local peaks and valleys of the $NAC(k)$ are then identified. For the detected peaks and valleys, their position and magnitude are recorded. Let $M$ be the number of peaks and $N$ be the number of valleys. Let $p\_posi(i)$, $p\_magn(i)$ $(i = 1, 2, \ldots, M)$ be the positions and magnitudes of these peak points, respectively, and let $v\_posi(j)$, $v\_magn(j)$ $(j = 1, 2, \ldots, N)$ be the positions and magnitudes of the valley points, respectively. The contrast of the projection is then defined to be

$$contrast = \frac{1}{M}\sum_{i=1}^{M} p\_magn(i) - \frac{1}{N}\sum_{j=1}^{N} v\_magn(i). \tag{9}$$

*Peak Analysis*: Given a peak sequence $p\_posi(i)$ including all the peaks detected form a projection and the number of peaks is $M$, the average of the distances among the successive peaks, *dis*, and the square root of the standard deviation of distances, *std* are computed. Let

$$\gamma = \frac{std}{dis}. \tag{10}$$

A lower variance in the distances between peaks implies a more "consistent" repetitive pattern. A threshold can then be set to distinguish between regular and irregular patterns. If $\gamma$ is smaller than a pre-selected threshold $T_p$, the corresponding projection is considered to represent a repetitive or

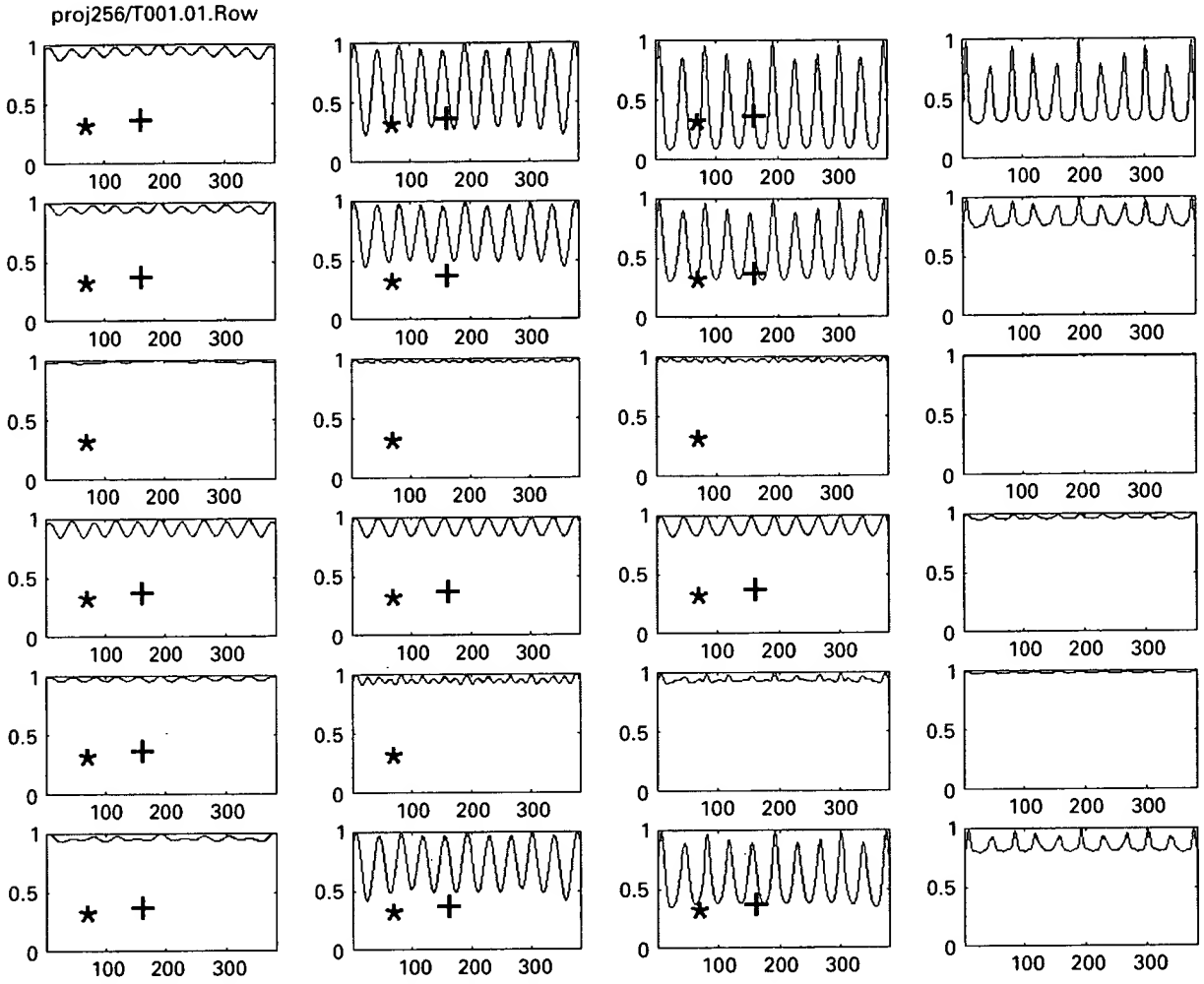



Fig. 3. *NAC* of horizontal projections of all the 4 × 6 filtered images from image T001.01. The projections labeled with '*' are the detected potential candidates and those also labeled with '+' are the final candidates after clustering.

regular pattern. Those projections that pass this threshold are then checked for consistency. A simple *agglomerative clustering* [5] in the two-dimensional *std–dis* space is then used to remove the outliers.

Fig. 3 shows the *NAC* of the 24 horizontal projections for the image T001.01 (shown in Fig. 2(a)). The projections marked with "*" are the ones that pass the threshold test. Fig. 4(a) shows the distribution of *std–dis* of these potential candidates. Fig. 4(b) shows the results after the clustering. Those projections that pass the consistency check are marked with a "+" in Fig. 3. A similar analysis is performed on the vertical projection as well.

From those projections that passed the consistency check, we identify the ones with the maximum contrast. Let $(m^*(H), n^*(H))$ denote the scale and orientation indices, respectively, of the horizontal projection with the maximum contrast. Similarly, let $(m^*(V), n^*(V))$ denote the scale and orientation, respectively, of the vertical projection with maximum contrast. Then, we have

$$PBC[v_4] = m^*(H) \quad \text{and} \quad PBC[v_2] = n^*(H),$$

$$PBC[v_5] = m^*(V) \quad \text{and} \quad PBC[v_3] = n^*(V).$$

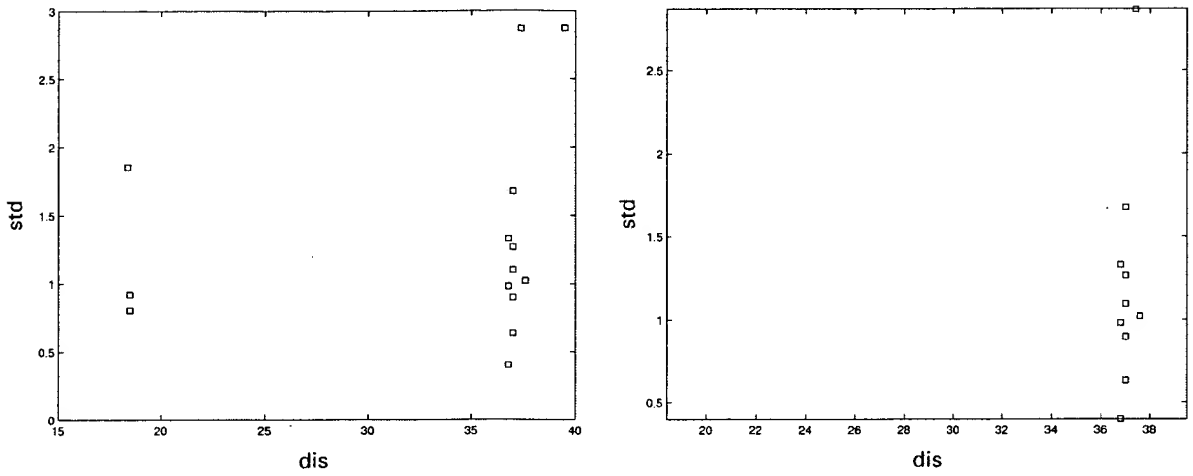



Fig. 4. Clustering of potential candidates: the left figure shows the distribution of potential candidates from the projections shown in Fig. 3 and the right one shows the final candidates after clustering.

### 3.2. Computing the Structuredness ($PCB[v_1]$)

The method of measuring the degree of the structuredness is based on the following observations on the distribution of candidate vectors.

- For strong structured textures, their periodicity could be captured by multiple projections – the candidates chosen from the above procedure. Typically, these candidates are neighbors in the scale-orientation space.
- If the texture is not structured or only weakly structured, the distribution of the candidates, if they exist, is usually sparse and the neighboring relationship can rarely be detected.

If such a consistency in the neighboring projections is detected from the projections in the candidate set, this would result in a larger credit, indicating a stronger structuredness. Based on these observations, the candidate projections are further classified as follows:

#### 3.2.1. Candidate classification

$C_1$: For a specific candidate, we can find at least one other candidate at its neighboring scale or orientation. The value associated with this class is $V_1 = 1.0$.

$C_2$: For a specific candidate, we can find at least one another candidate distributed at the same scale or orientation, but no candidate is located at its neighboring scale or orientation. The value associated with this class is $V_2 = 0.5$.

$C_3$: The candidate is the only one distributed at its scale and orientation. The value associated with this class is $V_3 = 0.2$.

At this stage, each of the candidate projections has an associated value computed based on the above classification. Let

$$M = \sum_{i=1}^{3} N_i * V_i, \tag{11}$$

where $N_i$ is the number of candidate projections classified as $C_i$. $M$ is calculated for the horizontal ($M_H$) and vertical ($M_V$) projections. Let

$$M_{img} = M_H + M_V \tag{12}$$

$M_{img}$ is quantized into $N_v$ bins by using *option decision tree classifier* [2]. The larger the value of $M_{img}$ is, the more structured the corresponding texture is. In our current implementation, $N_v = 4$. Consequently, each image is associated with a number $B_{img}$, $B_{img} \in \{1, \ldots, N_v\}$, to indicate which bin an image belongs to.

$PBC[v_1] = B_{img}$.

## 4. Extraction of similarity retrieval component (SRC)

### 4.1. Computing the similarity retrieval component (SRC)

The mean $\mu_{mn}$ and the standard deviation $\sigma_{mn}$ of the magnitude of the transform coefficients are used to form the *SRC*:

$$\mu_{mn} = \iint |W_{mn}(x, y)| \, dx \, dy \quad \text{and}$$

$$\sigma_{mn} = \sqrt{\iint (|W_{mn}(x, y)| - \mu_{mn})^2 \, dx \, dy}. \tag{13}$$

The similarity retrieval component (SRC) vector is now constructed using $\mu_{mn}$ and $\sigma_{mn}$. For $S$ scales and $K$ orientations, this results in a vector

$$SRC = [\mu_{11} \; \sigma_{11} \; \cdots \; \mu_{SK} \; \sigma_{SK}].$$

Note the double index on the vector elements. In the experiment, we use four scales $S = 4$ and six orientations $K = 6$, resulting in a feature vector

$$SRC = [\mu_{11} \; \sigma_{11} \; \cdots \; \mu_{46} \; \sigma_{46}]. \tag{14}$$

### 4.2. Distance measure for similarity retrieval component (SRC)

To perform the similarity retrieval, a distance measure is defined on the proposed feature vector. Consider two image patterns $i$ and $j$. Then the distance between the two patterns is defined to be

$$d(i, j) = \sum_m \sum_n d_{mn}(i, j), \tag{15}$$

where

$$d_{mn}(i, j) = \left| \frac{\mu_{mn}^{(i)} - \mu_{mn}^{(j)}}{\alpha(\mu_{mn})} \right| + \left| \frac{\sigma_{mn}^{(i)} - \sigma_{mn}^{(j)}}{\alpha(\sigma_{mn})} \right|, \tag{16}$$

$\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$ are the standard deviations of the respective features over the entire database, and are used to normalize the individual feature components.

## 5. Experiment results

### 5.1. Browsing using PBC

The parameters values used in the experiments are: $U_l = 0.04$, $U_h = 0.5$, $S = 4$, $K = 6$ (in Eqs. (3) and (4)) and $N_v = 4$. Thus, the resulting Gabor filter set has six orientations (30° intervals) and four scales.

The PBC vectors for some of the Brodatz texture images [1] are shown in Figs. 5 and 6. The size of the images in the original Brodatz album is 512 × 512. For evaluation purpose, each 512 × 512

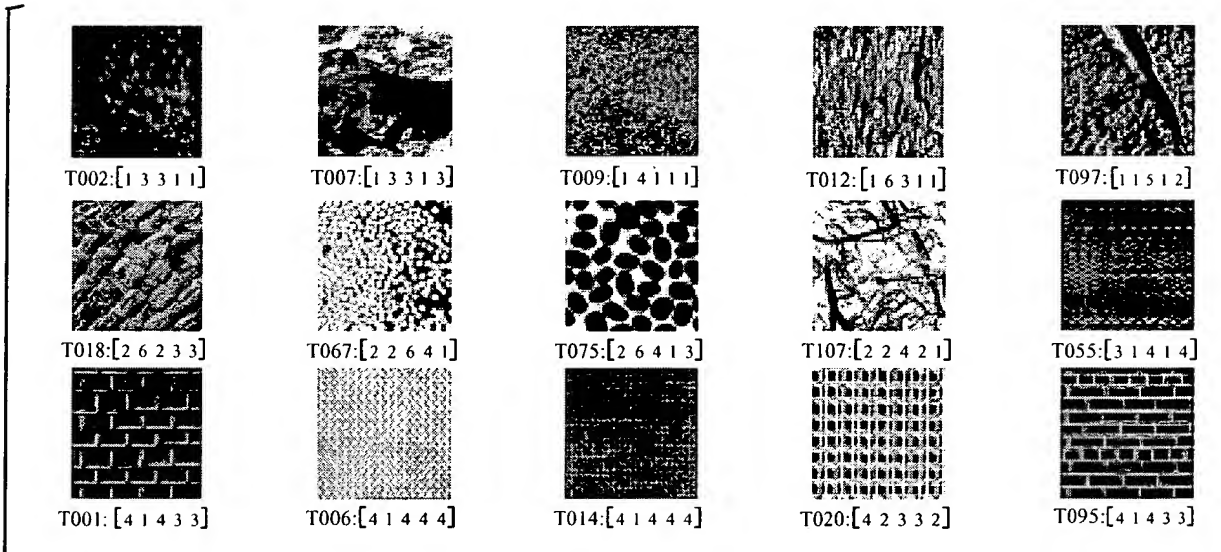



Fig. 5. PBC of some Brodatz textures.

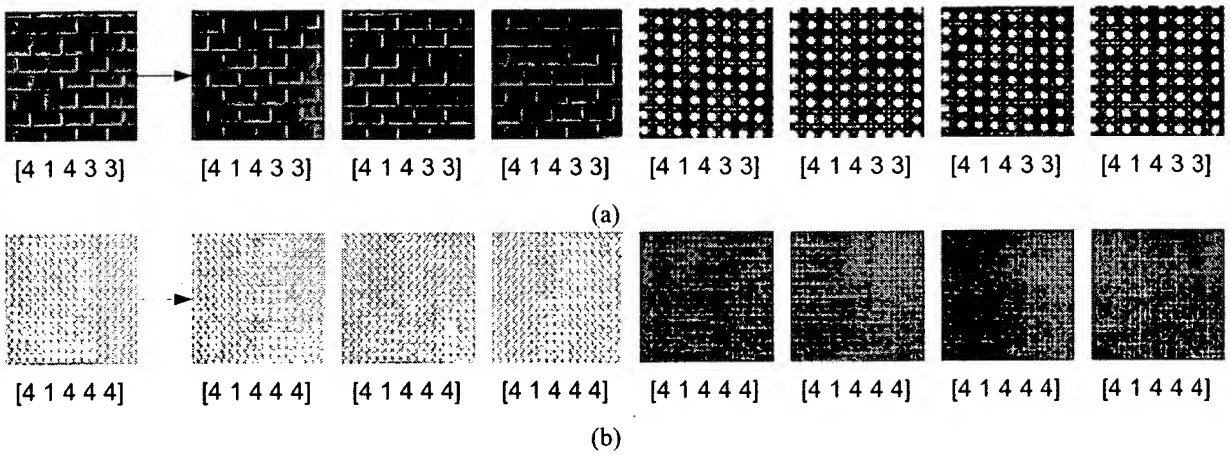



Fig. 6. Browsing example: patterns having similar PBC to the query pattern (on the left). The PBC values are shown below each texture.

image is divided into four $256 \times 256$ subimages. Each of the images shown in Fig. 5 is just one of the four subimages of each texture image. The $PBC[v_1]$ has values between 1 and 4 ($N_v = 4$). It could be observed that for the structured images, the estimated directions and scales match the perceived images very well. But the scale and direction estimates are not very reliable for textures with low values for $PBC[v_1]$.

The PBC computations are subjectively evaluated as follows. The 30 texture images from Fig. 5 were shown to five different individuals. They were asked to quantify the texture structuredness, directionality and scale on the same scale as our PBC computation. The median values of each of the components are used for comparing with the PBC values computed by our method. For the computer-generated PBC values, we use the median of the values from the four sub-images of each texture.

For the structuredness component $PBC[v_1]$, the computer and human generated values are within one value deviation for 28 of 30 images. If we consider values greater than or equal to 2 as representing the structured texture, the computed PBC values result in 17 structured and 13 non-structured textures. This is in good agreement with the human observers who agree with 16/17 (structured) and 12/13 (non-structured).

The computed dominant directions are also in good agreement with the human observers for the textures rated as structured. In 12 out of 16, the results are in complete agreement. It is observed that if a texture has horizontal and vertical patterns, the algorithm would pick up the corresponding diagonals as the directions. For the dominant scales, the human subjects had difficulty rating the textures on a scale of 4 and provided only one dominant scale for each pattern. It would have been more convenient, perhaps, to use the three scales – fine, medium and course – for the subjective tests. For the structured textures, the subjective and computed values for the first dominant scale were in agreement within one value deviation. Our proposed method did quite well in identifying scales for textures that had pattern at two significantly different scales. See, for example, T053 and T055 in Fig. 5, which contain pattern at different scales.

### 5.2. Similarity retrieval using SRC

In [13] we provided a comprehensive comparison with other state-of-art texture descriptors. The Brodatz texture album [1] is used in those experiments. This includes two descriptors based on orthogonal wavelets, SRC and [3], and one based on multiresolution simultaneous autoregressive model (MR-SAR) [14]. The SRC compares quite favorably with those other texture descriptors. The main observations from [13] are:

- In general, feature components corresponding to higher frequencies have better discriminating performance. However, decomposing the

high-frequency bands further in the tree-structured wavelet representation of [3] often leads to a decrease in performance, indicating that these features are not very robust.

- Experiments with different orthogonal wavelet transforms indicate very little variation in performance with respect to the choice of filters.
- The marginal improvement of the tree structured wavelet features comes at the expense of having a much larger feature vector, which adds to the overhead associated with indexing and searching.
- It is important to explore different similarity measures for each of the different sets of features. For example, using the Mahalanobis distance instead of the Euclidean distance improved the performance from 64% to 73% for the MR-SAR features. Normalized Euclidean distance worked better for all the others.

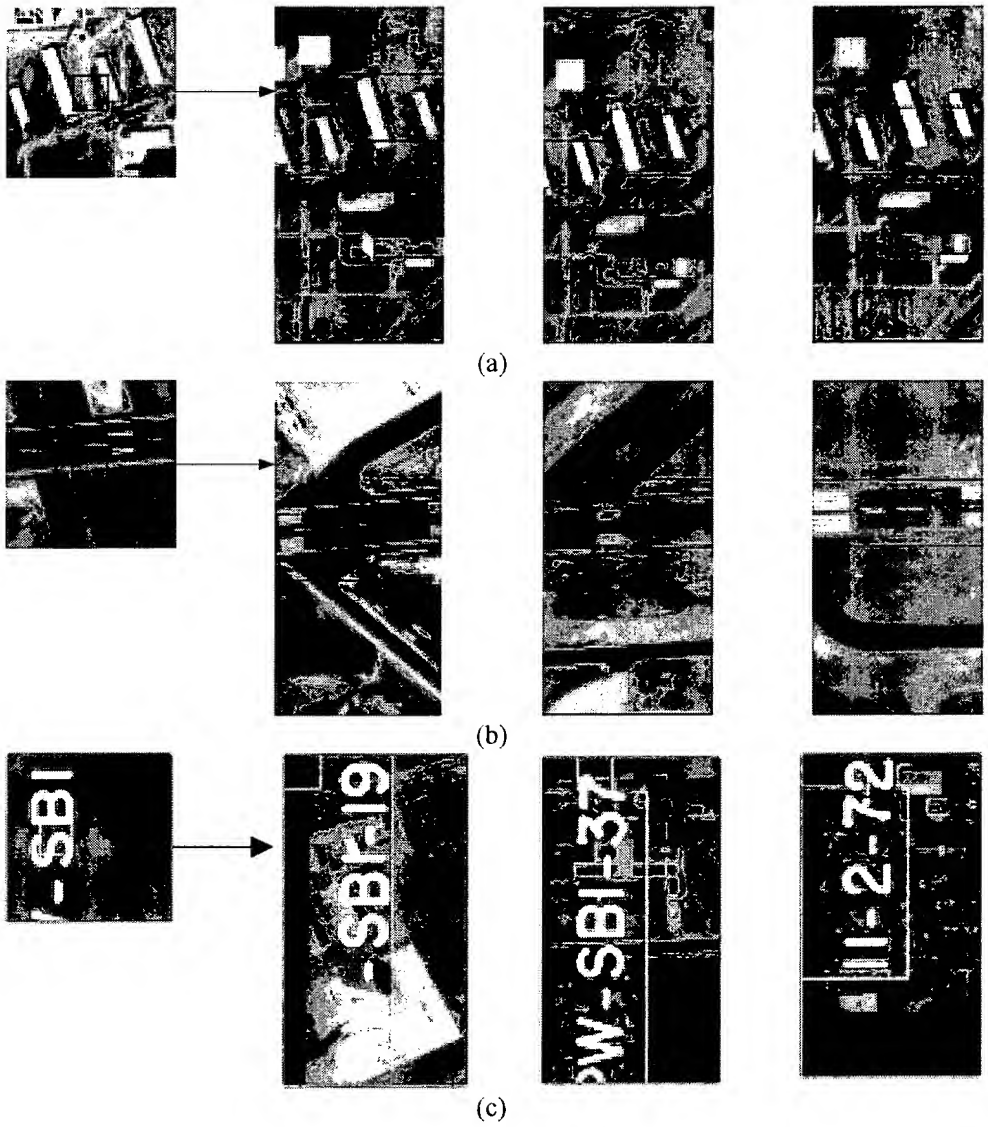



Fig. 7. Similarity retrieval using SRC on an airphoto database: (a) the region retrievals from areas containing some buildings; (b) an example of retrieving a part of the runway of an airport; and (c) retrievals containing an image identification number.

- For Brodatz images, the best results using the Gabor features were obtained using four scales and six orientations within each scale.

In [11], we provided an application to search and retrieve of aerial photographs using the SRC descriptor. Some retrieval examples on the airphoto database are shown in Fig. 7.

## 6. Discussions

We have presented a texture descriptor for browsing and similarity retrieval applications. A comprehensive evaluation of its performance in similarity retrieval is given in [13]. The browsing component extends its functionality, and enables coarse level classification of the database.

In the UCSB digital library project, the descriptor is used to facilitate query by example in a large aerial photograph database. The proposed texture descriptor provides a robust representation of many geographically salient features such as housing developments, parking lots, highways, airports, and agricultural regions. Details of this work can be found in [11].

The proposed descriptor has been used in other application domains as well. For example, in [8], researchers from IBM have reported applying this texture descriptor to an image database related to petroleum exploration. They concluded that the Gabor feature set outperforms other texture features (computed using the quadratic-mirror filter, the discrete cosine transform, and the orthogonal wavelet transform) by a wide margin on their benchmark dataset. This is consistent with our earlier observation.

## 7. Uncited Reference

[7]

## Acknowledgements

We would like to thank Dr. Wei-Ying Ma for his help in preparing this paper. S. Newsam is supported by an AASERT award from ONR #N00014-98-1-0515. This research is supported in part by Samsung Electronics and by a grant from NSF (award #97-04785).

## References

[1] P. Brodatz, Textures: A Photographic Album for Artists & Designers, Dover, New York, 1966.
[2] W. Buntine, Learning Classification Trees, Statist. Comput. 2 (2) (1992) 63–73.
[3] T. Chang, C.-C. Jay Kuo, Texture analysis and classification with tree-structured wavelet transform, IEEE Trans. Image Process. 2 (4) (October 1993) 429–441.
[4] J.G. Daugman, Complete discrete 2D Gabor transforms by neural networks for image analysis and compression, IEEE Trans. ASSP 36 (July 1988) 1,169–1,179.
[5] R.O. Duda, P.E. Hart, Pattern classification and scene analysis, Wiley, New York, 1970.
[6] W. Equitz, W. Niblack, Retrieving images from a database using texture-algorithms from the QBIC system, Technical Report RJ 9805, Computer science, IBM Research Report, May 1994.
[7] G.M. Haley, B.S. Manjunath, Rotation-invariant texture classification using a complete space-frequency model, IEEE Trans. Image Process. 8 (2) (February 1999) 255–269.
[8] C.S. Li, J.R. Smith, V. Castelli, L. Bergman, Comparing texture feature set for retrieving core images in petroleum applications, Proceedings of SPIE Storage and Retrieval for Image and Video Databases VII, SPIE Vol. 3656, San Jose, CA, January 26–29, 1999, pp. 2–11.
[9] A. Lindsay (Ed.), MPEG-7 Applications Document V. 9, ISO/IEC JTC1/SC29/WG11/Document #N2860, July 1999, Vancouver, Canada.
[10] F. Liu, R.W. Picard, Periodicity, directionality, and randomness: Wold features for image modeling and retrieval, MIT Media Lab Technical Report No. 320, March 1995.
[11] W.-Y. Ma, B.S. Manjunath, A texture thesaurus for browsing large aerial photographs, J. Amer. Soc. Inform. Sci. 49 (7) (1998) 633–648.
[12] W.Y. Ma, Hong Jiang Zhang, Benchmarking of image features for content-based retrieval, Proceedings of the 32nd Asilomar Conference on Signal, System & Computers, 1998.
[13] B.S. Manjunath, W.Y. Ma, Texture features for browsing and retrieval of image data, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 837–842.
[14] J. Mao, A. Jain, Texture classification and segmentation using multiresolution simultaneous autoregressive models, Pattern Recognition J 25 (2) (1992) 173–188.
[15] MPEG-7 Context, Objective and Technical Roadmap, ISO/IEC JTC1/SC29/WG11/Document #N2861, July 1999, Vancouver, Canada.
[16] F. Periera (Ed.), MPEG-7 Requirements Document V. 9, ISO/IEC JTC1/SC29/WG11/Document #N2859, July 1999, Vancouver, Canada.
[17] H. Tamura, S. Mori, T. Yamawaki, Texture features corresponding to visual perception, IEEE Trans. Systems Man Cybernet SMC8 (6) (1978) 00.
[18] P. Wu, W. Ma, B.S. Manjunath, H. Shin, Y. Choi, MPEG-7 Document, ISO/IEC JTC1/SC29/WG11/P77, February 1999, Lancaster.

**B.S. Manjunath** received the B.E. in Electronics (with distinction) from the Bangalore University in 1985, and M.E. (with distinction) in Systems Science and Automation from the Indian Institute of Science in 1987, and the Ph.D. degree in Electrical Engineering from the University of Southern California in 1991. He joined the ECE department at UCSB in 1991 where he is now an Associate Professor. During the summer of 1990, he worked at the IBM T.J. Watson Research Center at Yorktown Heights, NY. Dr. Manjunath was a recipient of the national merit scholarship (1978–85) and was awarded the university gold medal for the best graduating student in electronics engineering in 1985 from the Bangalore University. His current research interests include computer vision, learning algorithms, image/video databases and digital libraries. He is currently an Associate Editor of the IEEE Transactions on Image Processing and is a guest editor of a special issue on image and video processing for digital libraries to be published in the IEEE Image Processing Transactions in January 2000.

**Peng Wu** is a Ph.D. candidate in the Vision Research Laboratory at the University of California at Santa Barbara. He is currently working on indexing and searching image and video objects in large databases. His research interests include image/video analysis for content based retrieval and the management of large multimedia databases.

**Shawn Newsam** was born in Harare, Zimbabwe in 1968. He received the B.S. degree in Electrical Engineering and Computer Science from the University of California at Berkeley in 1991 and the M.S. degree in Electrical and Computer Engineering from the University of California at Davis in 1996. He is currently a Graduate Student Researcher in the Vision Research Laboratory at the University of California at Santa Barbara. His current research interests are in multi-media databases and digital libraries.

**Hyundoo Shin** received the B.S. degree in Applied Physics from Columbia University in 1983, the M.S. degree and the Ph.D. degree in Applied Mathematics from Brown University in 1990. From December 1990 to August 1992 he was a Postdoctoral Associate at Yale University, and then from September 1992 to January 1994 he was a Research Associate at Brown University. He joined SAMSUNG Electronics as a Senior Researcher in February 1994. He is currently a General Manager in the Infomedia Laboratory at SAMSUNG Electronics.